Application Note
0380-0264-10 Rev 00

Embedded Performance, Inc.
606 Valley Way, Milpitas, CA 95035
Telephone: (408) 957-0350
FAX: (408) 957-0307
e-mail: support@epitools.com
web: www.epitools.com

Using a MAJIC® Probe with Intel® SDT

0380-0264-10 Rev 00

April 30, 2004

Introduction

This application note describes how to install and use the EPI MAJIC® plug-in for the XDB debugger, part of the "Intel® C++ Software Development Tool Suite For Palm OS, Nucleus OS, Symbian OS (debugger only), and OS independent systems" (iSDT). It also supplements the iSDT documentation with information specific to using the iSDT debugger (XDB) with MAJIC® series of intelligent debug probes.

Additional Documentation

Additional documentation for the EPI Development Toolkit for ARM (EDT) software package and MAJIC® probe can be found in the `./manuals` directory of the EDT installation, and the `./manuals` directory on the EDT software package CD. The EDT Documentation Index (`edta_doc_index.html`) provides easy access to all the installed EDT documentation. It can be accessed via a shortcut in the Windows `Start` menu in the `Programs | EPI Tools – EDTA` program group.

All iSDT documentation can be found in the `./doc` directory of the SDT installation, and can be accessed via shortcuts in the Windows `Start` menu in `the Programs | Intel(R) C++ Software Development Tool Suite 1.2 -> Manuals` program group. The following EDT and iSDT documents will prove particularly useful through the course of this application note:

MAJIC User's Manual Complete information on configuration and operation of the MAJIC® probe, and the MON command language.

Using A MAJIC® Probe With Intel XScale® Microarchitecture
Covers configuration and operational details of the MAJIC® probe that are specific to targets based on Intel XScale® technology.

Intel® XDB Debugger Reference Manual
Documents the common user interface and command language shared by all versions of the XDB debugger.

Intel® XDB JTAG Debugger User's Manual
Supplements the reference manual with a tutorial and information specific to using XDB with a parallel port JTAG probe.

Intel® XFlash User's Manual Documents the `xflash` command line utility used to support automated flash programming in a production environment.

In the Intel® documentation, the term "JTAG" is not used in a generic sense. "JTAG" refers to a specific non-EPI JTAG probe and "XDB JTAG" refers to the XDB debugger configured to load the plug-in that interfaces to that probe. We use the term "XDB-MAJIC" to refer to the XDB debugger configured to load the EPI plug-in that interfaces to the MAJIC probe.

Getting Support

Please do not hesitate to contact our technical support group if you have any questions or need assistance in configuring the MAJIC® probe for your system. We recognize that these issues are complex, and are committed to making sure our tools work well for you.

Installation

Installation and configuration consists of the following steps:

1. Install the iSDT package. Insert the Version 1.2 iSDT CD into your CD-ROM drive. Your web browser should start up automatically, displaying the CD home page. If the browser installer does not launch after a few seconds, then use the Windows [`Start`] menu, select `Run...` and [`Browse`] to find your CD drive, then run `index.htm.` Click the "Install Tool Suite" link to launch the iSDT installation program, and follow the prompts to complete the installation.

   Note: The installer will ask for a license file, which you must get from the Intel® Registration Center before you can complete the installation. Please refer to the *Registration Instructions* that were provided with your iSDT package for the software serial number, and instructions for requesting the license file.

2. If you have the Metrowerks CodeWarrior IDE already installed on your PC, and you wish to use it with the iSDT compiler and tools, you can then click the "Install Intel® C++ Software Development Tool Suite Plug-in For Metrowerks* CodeWarrior*" link to integrate the iSDT Tool Suite with CodeWarrior.

3. Install the EDT package. Insert the EDT CD into your CD-ROM drive. If the installer does not launch after a few seconds, then use the Windows [`Start`] menu, select `Run...` and [`Browse`] to find your CD drive, then run `setup.exe`. Follow the instructions provided by the installer.

4. Integrate the MAJIC® target plug-in with the XDB debugger in the iSDT package. This step adds a target named "MAJIC XScale COM1" that can be selected from the XDB Settings dialog, configured by default to connect to the MAJIC® probe using the COM1 serial port, with PXA250 as the CPU type and standard initialization files for the DBPXA250 target board. It also adds shortcuts to the iSDT "Examples" and "Configurations" program groups to invoke XDB-MAJIC™ for the evaluation boards and example projects that iSDT supports out-of-the-box, with corresponding CPU type and board initialization files.

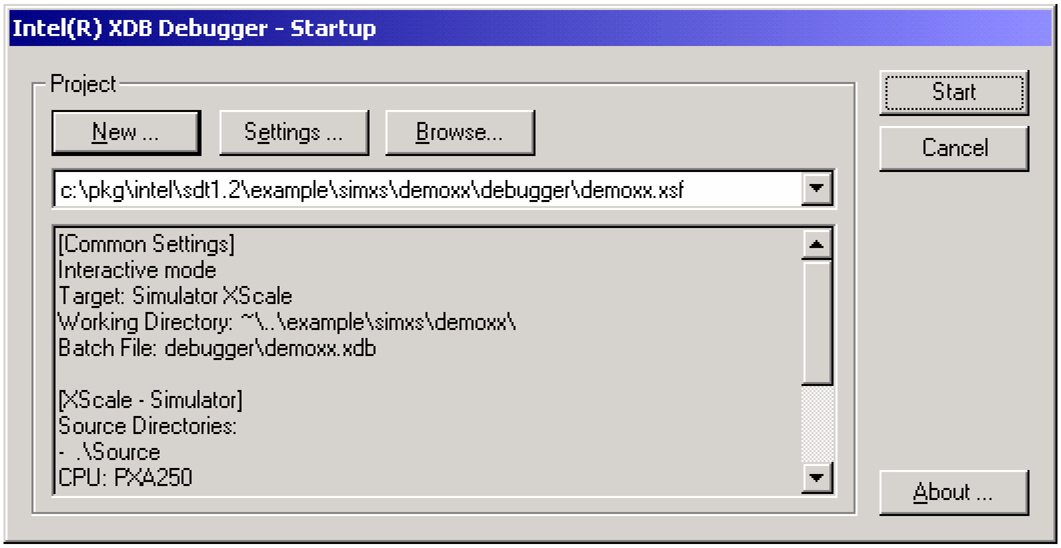
   If iSDT was installed before EDT, the EDT installer should detect the installed iSDT package and offer to install the plug-in for you. However, if you installed EDT first, or if you declined to install the plug-in when you installed EDT, you can manually run the plug-in installer after both EDT and iSDT have been installed. Use the Windows [`Start`] menu, select `Run...` and [`Browse`] to find the `./udi` subdirectory in the directory where you installed the EDT package, and run `isdt_install.exe`.

5. Configure default settings for XDB-MAJIC. This step updates the default MAJIC® configurations created in step 3 to reflect your own target system and your MAJIC® probe's hostname or IP address for the preferred Ethernet connection. It adds a target named "MAJIC XScale Connection" that can be selected from the XDB Settings dialog, configured to connect to the MAJIC probe using the connection method, processor type, and initialization file that you specify. These defaults provide the initial Settings values whenever you create a new XDB project using the "MAJIC XScale Connection" target.

   Please refer to the *MAJIC User's Manual* for instructions on configuring a MAJIC® probe for your target system, especially Chapter 3 *Debug Environment*. The first section describes using the MAJIC Setup Wizard. In the "Choose Operation" step, you should select "XDB for MAJIC" in the "Choose your debugger:" drop-down list . After you click [Go], select "Setup default XDB MAJIC connection settings" in the next dialog, then continue along as described in the MAJIC User's Manual. In the "Perform Setup" form, the available actions are "Setup Default XDB MAJIC Settings" and "Setup XFlash Utility's MAJIC Settings". Both should be enabled. The first action creates or updates the "MAJIC XScale Connection" target and updates the various Example and Configurations XDB project settings. The second updates the settings for the MAJIC® probe's version of the XFlash command line utility in the `xflash_majic` directory.

Using XDB with a MAJIC® Probe

For the most part, using XDB with a MAJIC® probe is the same as using XDB with any other type of target. When you launch XDB, you get a "Startup" dialog where you can select an existing XDB "Settings" file, click <Settings> modify the settings loaded from the file, or click <New> to create a new Settings file.

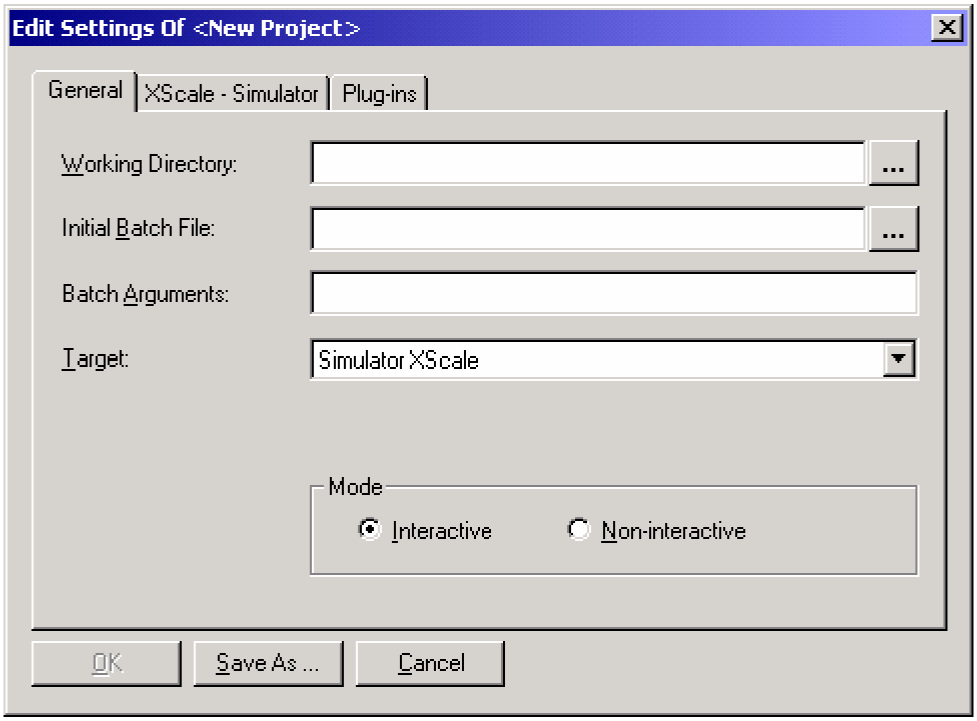


c:\pkg\intel\sdt1.2\example\simxs\demoxx\debugger\demoxx.xsf

```
[Common Settings]
Interactive mode
Target: Simulator XScale
Working Directory: ~\..\example\simxs\demoxx\
Batch File: debugger\demoxx.xdb

[XScale - Simulator]
Source Directories:
- .\Source
CPU: PXA250
```

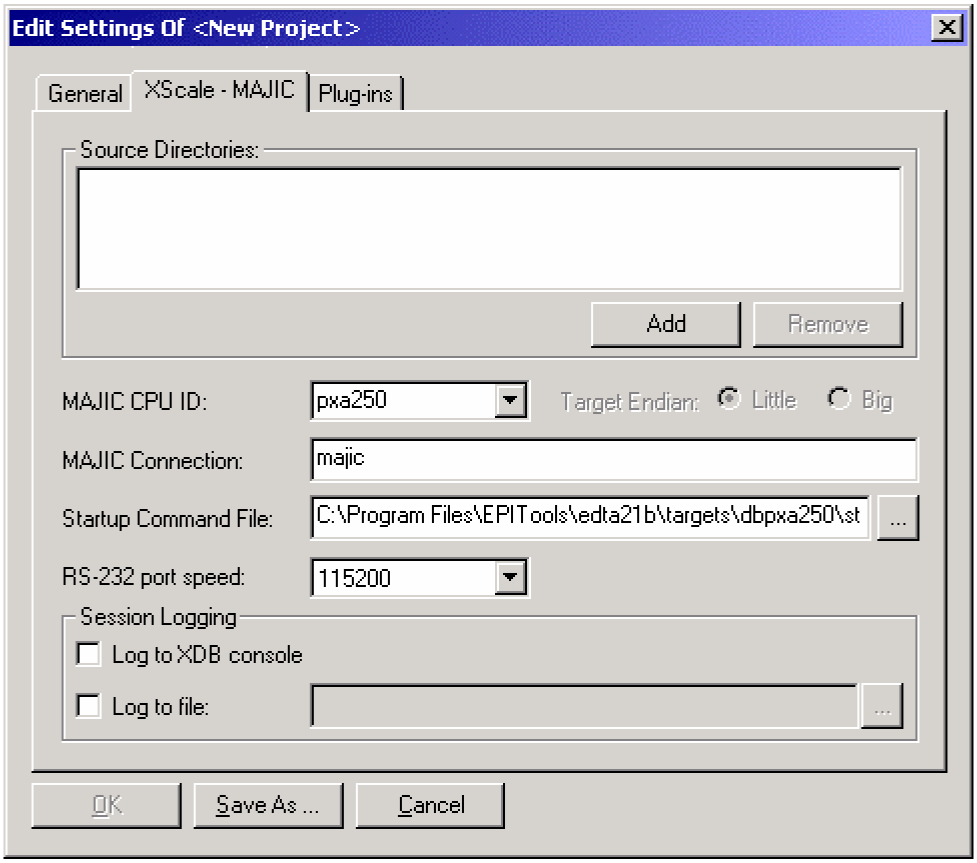
When you click <Settings> or <New>, you get the XDB "Edit Settings" dialog below.



Target: Simulator XScale

Mode: ● Interactive ○ Non-interactive

The "Target:" drop-down box allows you to select any supported target type, and the middle tab will change to reflect the selected type. When you select "MAJIC XScale COM1" or "MAJIC XScale Connection", the middle tab changes to "XScale – MAJIC". Clicking on the "XScale – MAJIC" tab displays the following dialog, where you can edit the XDB settings that are specific to the MAJIC® probe :



Source Directories:

MAJIC CPU ID: pxa250

Target Endian: ● Little ○ Big

MAJIC Connection: majic

Startup Command File: C:\Program Files\EPITools\edta21b\targets\dbpxa250\st

RS-232 port speed: 115200

Session Logging

□ Log to XDB console

□ Log to file:

Source Directories: Enter the pathnames of directories to be searched for source files for you project, if needed. This field is actually common to all XDB target types and is described in section 2.2.2.2 of the *Intel® XDB Debugger Reference Manual*".

The rest of the fields are MAJIC® probe specific:

MAJIC CPU ID: Select the correct EPI CPU identifier from the drop-down list. Not all Intel® XScale® cores supported by MAJIC® probes are included in the list, only those fully supported by XDB will appear.

Target Endian: Only little-endian targets are supported by XDB at this time.

MAJIC Connection: Enter `com1` or `com2` to specify a serial port to use for communication with the MAJIC® probe. When using Ethernet, enter the hostname or IP address assigned to the MAJIC® probe. It is not necessary to append ":e" to the hostname or IP address for Ethernet connections.

Startup Command File: Enter the pathname of the startup command file to initialize the MAJIC® probe and target board. This is the same command file you would use with EPI's MONICE and EDBICE debuggers, and it is processed by the MAJIC® plug-in when XDB first connects. This is documented in Chapter 3 of the *MAJIC® User's Manual*. Don't confuse this with the XDB startup command file you can specify in the General page, as described in section 2.2.2.1 of the *Intel® XDB Debugger Reference Manual*".

RS-232 port speed: This drop-down list controls the baud rate if you are using a serial connection. If you are not using the serial port to communicate with the MAJIC® probe, it doesn't matter what value is selected. The default value is 9600 baud.

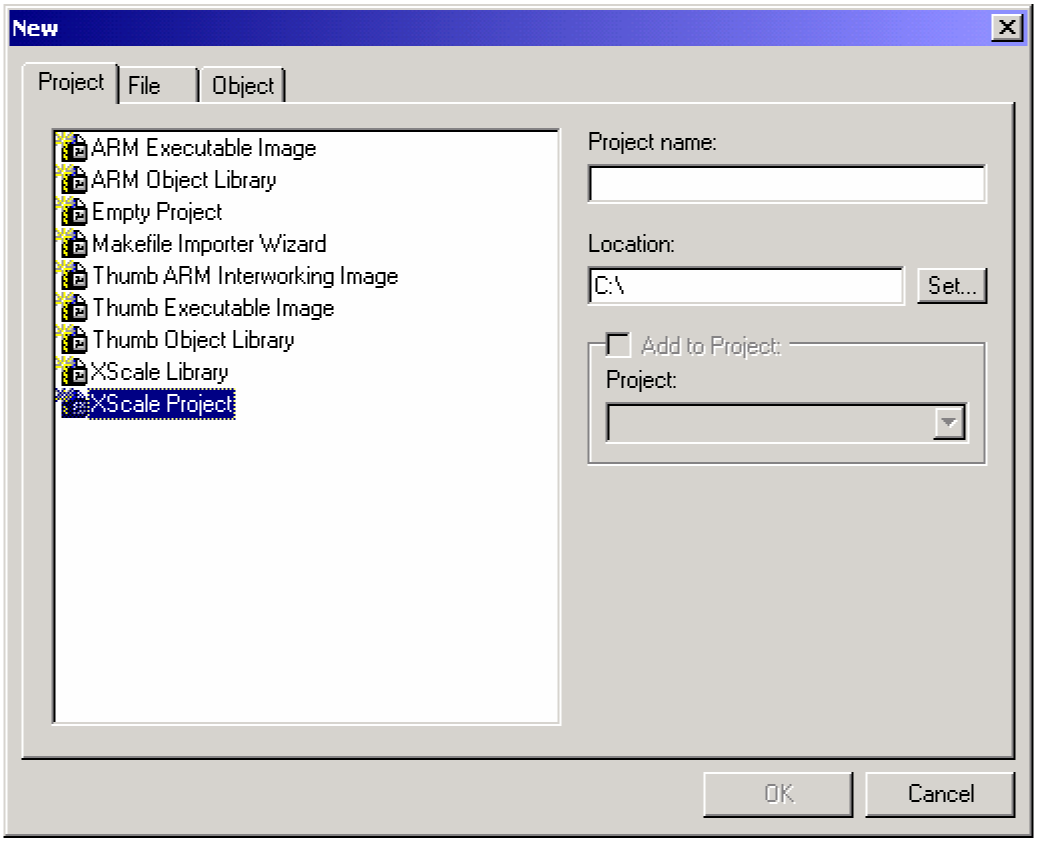
Session Logging: These options can be used to generate a detailed log of the API calls that XDB is making to the MAJIC® plug-in. If you are having a problem using XDB with your MAJIC® probe, EPI customer support may direct you to enable session logging to a file to help in diagnosing the problem. Otherwise these options should normally be disabled.

Any changes you make to your settings for an existing XDB Settings File are automatically saved when you click <OK>. If you are creating a new project, you will need to click <Save As…> and specify a destination directory and file name for the settings file. Normally you would save the file in the same directory as the program you are debugging.

Once your settings are correct (which they will be by default if you have followed installation step 4 above), power on your target system and MAJIC® probe and click <Start> to open XDB and connect to your MAJIC and target. Once in XDB, there are very few MAJIC® probe-specific issues for you to be aware of. They are described in the next section. If you are new to XDB, we recommend reading the Intel® manuals mentioned above and following the tutorial in Chapter 3 of the *Intel® XDB JTAG Debugger User's Manual*.

Using CodeWarrior® IDE with a MAJIC® Probe

Installing the Intel® C++ Software Development Tool Suite Plug-in For Metrowerks CodeWarrior plug-in adds two templates to the list you can choose from when creating a new project in the CodeWarrior® IDE:



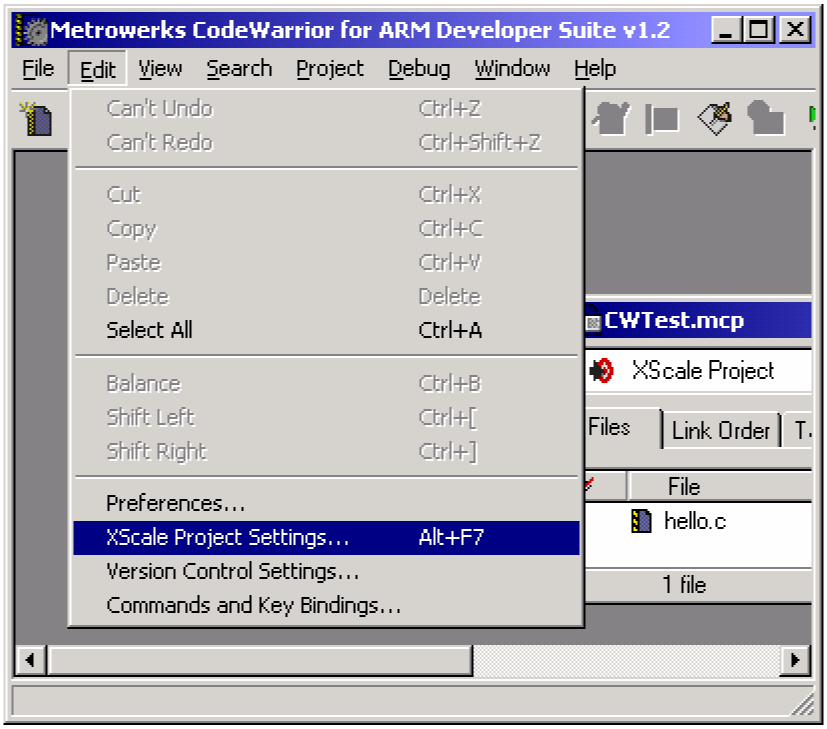
- ARM Executable Image
- ARM Object Library
- Empty Project
- Makefile Importer Wizard
- Thumb ARM Interworking Image
- Thumb Executable Image
- Thumb Object Library
- XScale Library
- XScale Project

Project name:

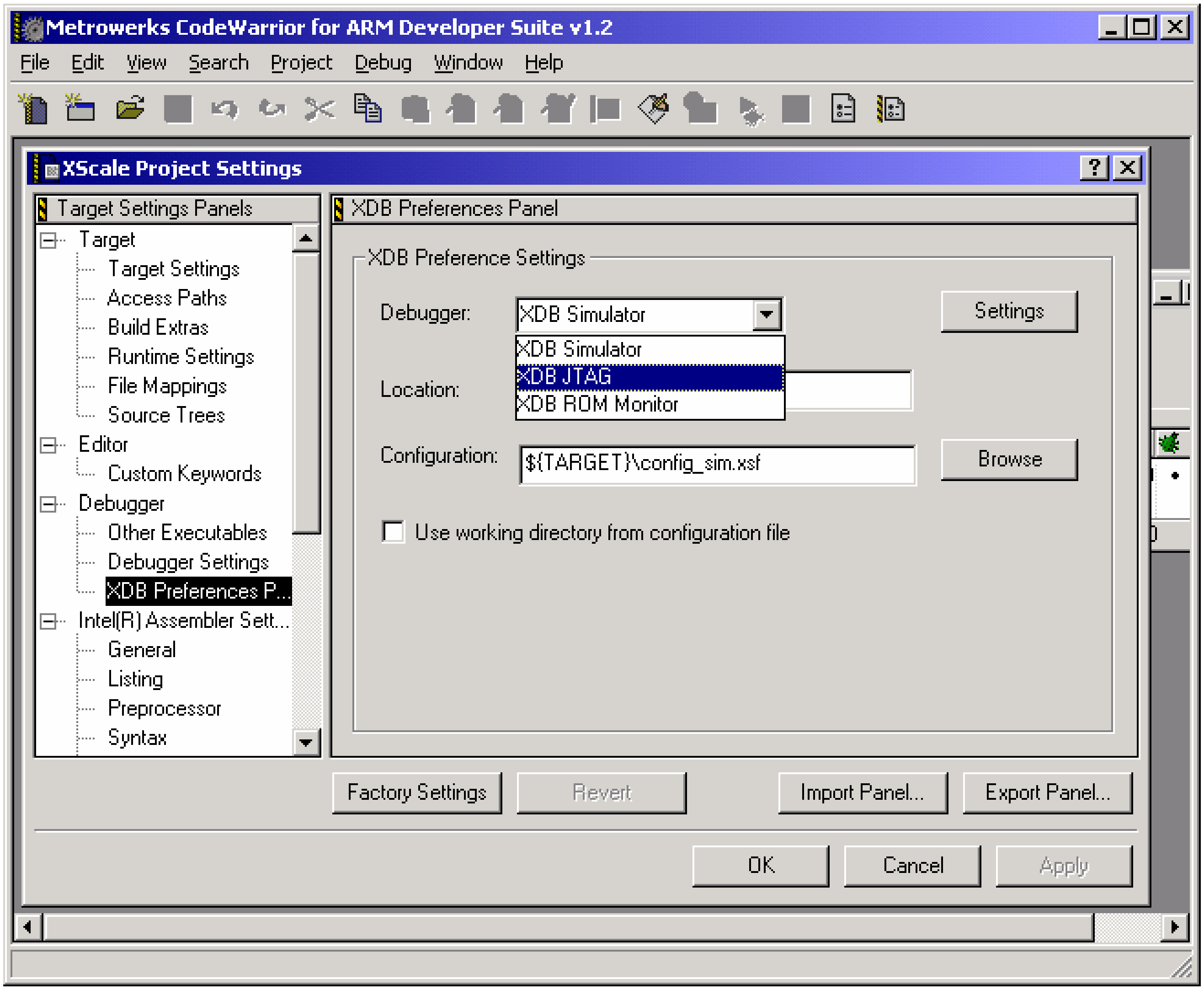
Location: C:\

Project:

“XScale Library” is a template for building a library using the iSDT compiler, and “XScale Project” is a template for building and debugging an application program using the iSDT compiler and XDB debugger. Creating a project using either template adds an iSDT-specific entry to the Edit menu for editing the project settings:



The XScale Project Settings dialog adds panels for configuring the appropriate iSDT tools to the standard CodeWarrior preference panels. The “XScale Library” template adds panels for the iSDT compiler, assembler, file converter and librarian. Similarly, the “XScale Project” template adds panels for the iSDT compiler, assembler, file converter, linker and debugger. To configure a CodeWarrior project based on the “XScale Project” template to use XDB-MAJIC for debugging, first select “XDB Preferences Panel” in the Target Settings Panels tree:



Debugger: XDB Simulator

Location:

Configuration: ${TARGET}\config_sim.xsf

☐ Use working directory from configuration file

Note that the "XDB Preference Settings" panel offers only three choices in the Debugger: drop-down list, as shown above. Unfortunately, it is not currently possible to have an "XDB MAJIC" choice added to this list. But you can alter the settings for any of the existing choices to use the MAJIC plug-in. When you click the `[Settings]` button next to the Debugger: drop-down list, the standard XDB "Edit Settings" dialog (as shown on pages 3 and 4) is opened. You can then select a MAJIC target type and configure the XDB-MAJIC settings as described in the "Using XDB with a MAJIC® Probe" section above. Your settings will be saved in the `.xsf` file specified in the Configuration: field of the XDB Preferences panel.

Corrections to XDB Manuals

Intel® XDB Debugger Reference Manual

Since the *Intel® XDB Debugger Reference Manual* describes the bulk of the XDB user interface which is independent of the a trget type used, it is almost completely applicable to XDB-MAJIC™. However, there is one undocumented feature and three issues relating to the MAJIC® probe to be aware of:

1. XDB supports a global startup command file in addition to the project-specific startup command file that can be specified in the General page of the Settings dialog (“Initial Batch file”), and the application-specific command file that can be specified in the Load dialog (“Initfile”). If there is a file named `startup.xdb` in the `SDT\xdb` directory, XDB will execute the commands in it every time it starts up.
2. Section 2.2.2.3 documents the “Target Connection: setting which is specific to the ROM-MONITOR and JTAG Target types. The corresponding MAJIC® option is labeled “MAJIC Connection:” to avoid confusion, and is described above.
3. The `CUSTOM` command documented in section 4.2.9 is applicable to a MAJIC® probe. The MAJIC® plug-in includes a MONICE-compatible command interpreter. Nearly all the MONICE commands described in the *MAJIC User’s Manual* are available, with the notable exceptions being breakpoint and execution commands. For example:

   ```
   xdb> custom "h"
   xdb> custom "h d"
   xdb> custom "h dv"
   xdb> custom "l c:\\myproject\\myprogram.axf"
   xdb> custom "dv \"dv is a MON command\n\" ; di"
   ```

   The first three example show how to enter the MON Help command for varying levels of help. The fourth example shows how to enter the MON Load command to download your program, which will give the fastest possible download time. The last example shows how to enter a command string consisting of a **`dv`** command followed by a **`di`** command. Note the need for backslashes to escape the backslashes in the program pathname and the quote characters that are part of the **`dv`** command, since the `CUSTOM` command requires the entire command line to be given as a quoted string. The syntax of quoted strings in XDB is the same as for the C language, so embedded quote and backslash characters need to be “escaped” by a backslash. In file pathnames, you can also use forward slashes rather than escaped backslashes.
4. Sections 4.2.11.7, 4.2.38.8, and 4.2.39.12 document the `DELETE`, `SET`, and `SHOW EXCEPTION` commands. `DELETE` and `SET` work as described, but `SHOW EXCEPTION` can display incorrect results.

   The MAJIC® probe has a configuration option named `vector_catch` that controls which exceptions cause execution to halt and which are ignored. This is typically set in the start-up command file selected by the MAJIC® Setup Wizard, or the XDB Edit Settings dialog shown above. The `SET EXCEPTION` and `DELETE EXCEPTION` commands call an API function that correctly updates the MAJIC® probe’s `vector_catch` option, so they work as expected. However, there is no corresponding query function in the API. `SHOW EXCEPTION` just assumes that the initial state is to ignore all exceptions, and keeps track of the changes made via `SET EXCEPTION` and `DELETE EXCEPTION`. If the `vector_catch` setting in your start-up file is set to catch any vectors, or if you change `vector_catch` directly with the CUSTOM command, the actual `vector_catch` setting will not be reflected by XDB’s `SHOW EXCEPTION` command.

Intel® XDB JTAG Debugger User's Manual

There are actually very few differences from XDB's point of view between the JTAG and MAJIC® target types, so this manual is also largely applicable to XDB-MAJIC if you simply read "MAJIC" everywhere the manual says "JTAG". Specific exceptions are listed below.

Sections 1.2 and 1.3, a description of the parallel port adapter used for XDB-JTAG and its restrictions, are not applicable. Chapter 2, Installation, is also not applicable to the MAJIC® probe and is superceded by this document.

You can follow the tutorial in Chapter 3 with XDB-MAJIC. Just use the "`MAJIC Debugger (`*`board_name`*`)`" Start menu shortcut rather than "`JTAG Debugger (`*`board_name`*`)`" or browse to `<installation-directory>\example\majicxs\demoxx\debugger` rather than `<installation-directory>\example\ocdxs\demoxx\debugger` to invoke XDB-MAJIC™ in section 3.1.1.

Most of Chapter 4 is also applicable to XDB-MAJIC. In section 4.1.1 (bullet 3), you should navigate to directory `<installation-directory>/xdb/configrations/majic` rather than `<installation-directory>/xdb/configrations/ocd`. In section 4.1.2, bullet 3, the Target for XDB-MAJIC™ Debugger is "MAJIC XScale Connection" or "MAJIC XScale COM1". In bullets 4 and 5, the example directories would be under `example\majicxs` rather than `example\ocdxs`. Bullets 8 and 10 through 13 are not applicable, refer to the description of the XScale – MAJIC settings page above.

Sections 4.3, 5.5, and 5.6 are not applicable to XDB-MAJIC The SWI Options page will be present, but any values entered in it or via the `SWI "SET OPTION"` command will not be used. With MAJIC, the `top_of_memory` configuration option is used to set the stack and heap limits. This option is typically set in the start-up command file selected by the MAJIC Setup Wizard, or the XDB Edit Settings dialog shown above.

In section 4.5, "Enhanced Breakpoints", you can not specify a specific breakpoint register for hardware breakpoints. The MAJIC® probe manages the breakpoint registers automatically.

Chapter 6, Diagnostic Messages, is specific to the JTAG Target and is not applicable to the MAJIC® probe. XDB-MAJIC™ displays error messages in the XDB Console window rather than relying on numeric codes.

Intel® XFlash User's Manual

The `–tgttype` option documented in the manual does not work. In fact, `xflash.exe` does not support multiple target types in any way. In order to provide a version of XFlash that works with the MAJIC® probe, `isdt_install.exe` creates a second copy of the XFlash utility in a sub-directory named `./xflash_majic`. In order for this copy to work correctly with your MAJIC® probe and target board, you need to use the EPI Setup Wizard to customize your default settings as described in the *Installation* section, step 5.

Other Issues

If you are new to XDB, you may be surprised to learn that XDB does not automatically set the PC to your program's entry point address when it is loaded. By default, execution will begin at the reset address regardless of where your program is loaded. The most convenient way to set the PC is to specify an "Initfile" (XDB batch file) in the Load dialog, and include in it a command to set the PC to whatever label is used to identify the entry point in your build environment. For example, the EPI samples directory contains a file named `set_pc.xdb` which includes the command:

```
set sysreg pc=&__main
```

When loading an EPI sample program, click the Initfile checkbox and browse to `set_pc.xdb`. Another approach to this issue is demonstrated in the iSDT examples. They use the"Initial Batch file" (`demoxx.xdb`), specified in the XDB settings and executed once when XDB starts up, to define toolbar buttons for "Download", "Reload", and "Start" that load the demo program and set the PC in one operation.

There is a problem with the Trace window if execution stops outside the bounds of the program being debugged, for example if clicking <Stop> happens to interrupt the program in an exception handler. If the Trace window is open, it does not display any data and XDB goes into an infinite loop fetching the PC and appears hung. Clicking the <Stop> toolbar button twice breaks the cycle and allows normal XDB operation to resume. The work-around for this issue is to keep the Trace window closed normally, and open it as needed.

There is a problem with the Output window used to display program “semi-hosting” output. If any register windows are open while the target program is running, they will be updated every time output is added to the Output window. This can cause a significant slowdown in execution performance as well as other problems. If your program is using semi-hosting to display output in XDB, we strongly recommend that you open register windows only when stopped at a breakpoint.